SAND20XX-XXXXR
LDRD PROJECT NUMBER: 166140
LDRD PROJECT TITLE: The Adaptive Multi-scale Simulation Infrastructure

PROJECT TEAM MEMBERS: William R. Tobin, Scientific Computation Research Center, Rensselaer Polytechnic Institute

ABSTRACT: The Adaptive Multi-scale Simulation Infrastructure (AMSI) is a set of libraries and tools developed to support the development, implementation, and execution of general multi-model simulations. Using a minimal set of simulation meta-data AMSI allows for minimally intrusive work to adapt existent single-scale simulations for use in multi-scale simulations. Support for dynamic runtime operations such as single- and multi-scale adaptive properties is a key focus of AMSI. Particular focus has been spent on the development on scale-sensitive load balancing operations to allow single-scale simulations incorporated into a multi-scale simulation using AMSI to use standard load-balancing operations without affecting the integrity of the overall multi-scale simulation.

INTRODUCTION: Most physical models and implementations using numerical methods operate at a single scale characteristic to the problem – the scale at which the preponderance of attributes governing the underlying physics being modeled are measured – and the scale at which the resolution of the solution is considered commensurate with the desired accuracy of the simulation.

Introduction of influence from scales orders of magnitude removed from the primary scale can take place in several different ways. This influence can be incorporated during the mathematical development of the physical model itself. In this case the multi-scale features are incorporated into the numerical implementation automatically, as a result of their presence in the mathematical model. Multi-scale influences can also be incorporated in a more strictly numerical sense, combining physical models operating at different scales but opera ting on some of the same physical fields and values [12].

Numerically multi-scale simulations associate many numerical implementations of single-scale physical models and cause them to interact in a meaningful way by passing key physical terms from some scales and supplying them to others, typically this will involve up-/down-scaling equations to take into account the difference in scale. This is only useful in cases where terms from the various incorporated scales contribute meaningfully to the solution at the primary scale of interest. The domain at the scale of interest (for brevity the engineering scale) is sufficiently large that conducting the entire simulation with a granularity orders of magnitude smaller would make the simulation computationally intractable.

This restriction requires the development of multi-scale simulation paradigms: algorithms and

operations which allow the influence of various far-removed scales to be accurately reflected in the simulation of the engineering scale. While there are several well-established multi-scale paradigms (particularly the heterogeneous multi-scale method as well as the concurrent multi-scale method), this is a rapidly advancing field and as new problems are considered additional multi-scale models will undoubtedly be developed. As such developing a set of tools to optimize a single multi-scale paradigm will be less impactful then the development of general tools which can be used themselves to implement new multi-scale paradigms, while supporting extant paradigms as efficiently as possible.

DETAILED DESCRIPTION OF EXPERIMENT/METHOD: Currently AMSI provides interfaces for describing and managing the execution of individual scales in a multi-scale simulation, for defining and managing the scale-coupling communication required by such simulations, and for planning and enacting scale-sensitive load balancing operations on individual scales in a multi-scale system.

AMSI operates by maintaining a set of minimal simulation metadata in order to model various quantities of interest during simulation execution. A decentralized approach toward control is taken; dynamic control decisions are made and implemented at runtime during collective operations. In order to avoid the introduction of unnecessary parallel barriers into the code, AMSI control decisions are only made during operations which are already collective over the set of processes effected by a given control decision.

During simulation initialization simulation scales are declared and then defined by associated them with process sets, and declaring their scale-linking relations. Process sets are simply mathematical sets of process ranks implemented so as to take advantage of any mathematical conveniences to minimize explicit storage whenever possible. At present process sets must be non-overlapping to take advantage of the simulation control portions of the AMSI libraries, but for the simulation metadata modeling, scale-coupling communication, and multi-scale load balancing operations, the process sets associated with related scales may overlap. Scales are declared to be related if some quantities of interest (such as tensor field values) are transmitted between them for scale-coupling.

By using data structures describing the dynamic state of the parallel execution space AMSI can determine which scales will need to interact, and only for these interacting scales are the required internal buffers and scale-coupling structures allocated. This approach is taken in order to reduce unneeded overhead caused by maintaining data structures specific to interacting scale-tasks, which would be the case when assuming that any two scale-tasks may interact. Dynamic redefinition of which scales interact is not currently possible, but will be implemented to support dynamic scale instantiation.

Scale-linking quantities are dynamically registered with the AMSI system by the scale-task that produces the data, along with information about which interacting scale-tasks will receive how much data. This data is stored in AMSI, and when scale-linking communication is to take place, a user- configurable algorithm operating on a mapping of all pertinent scale-linking data is used to provide a full plan of scale-linking communication. This plan is the disseminated (only partially) to those processes that need to know their individual role in the communication. At this

point scale-linking communication of any POD datatypes may take place at any time. If the quantity of scale-linking data changes at any point the previous steps must be repeated, but if the quantity remains the same for several instances of inter-scale communication the processes do not need to occur every time. Quantity here is a fairly abstract term, as the discrete units modeled by AMSI metadata structures is arbitrary: they can be any discrete unit of data that can be handled by the underlying communications system (MPI in most cases).

Multi-scale load balancing processes also make use of the meta-data structures concerning the discretization of the parallel execution space. AMSI currently has separate functionalities for: data removal, data addition with load balancing, and scale-sensitive load-balancing (migration). The removal routine does not automatically call any load balancing or migration routines and should normally be paired with the migration routine. In the current hierarchical multi-scale test problem these routines are called in the previously listed order. Data is first removed which may unbalance the workloads. Next, data is added with load balancing so that differences in workload can be filled by the new data. Finally, scale-sensitive load-balancing is used to rectify any remaining unbalance, though as the computational demand (weight) associated with the newly distributed data is typically not known a-priori, some semi- accurate weighting metric should be used to ensure the new data is not treated incorrectly. Currently each of these routines also assumes implicit ordering of data within AMSI since explicit ordering has not yet been implemented. This means that when using the load balancing and migration routines, the user's understanding of the data outside of AMSI needs to be reordered to maintain agreement across scales. This is a side-effect from the use of implicit (user-defined) ordering of scale-coupling data, which has the benefit of not introducing additional memory requirements to explicitly and uniquely number these pieces of data inside of AMSI. Further, AMSI provides the capability to automatically reorder the user data and provide a report on the updated ordering. Still as this may be undesirable in some cases, work is being conducted to provide an alternative option using unique numbering internal to AMSI to avoid effecting user data.

Due to the necessity of retaining inter-scale linkages through the load-balancing processes, the actual data migration undertaken to implement a specific load balancing plan is typically conducted internally by the AMSI system that is provided with buffers of data being load-balanced by each process on the appropriate scale. However, in order to maintain usability in more general situations, it is also possible to simply inform the AMSI system that the current load-balancing plan has already been accomplished by user-implemented data migration. This functionality combined with the ability to use user-designed load-balance planning algorithms allows AMSI to work with entirely external load-balancing libraries and algorithms, allowing scale-sensitive load-balancing to be used even when a specific multi-scale use case falls outside the bounds of capabilities built into the system.

RESULTS AND DISCUSSION: A multi-scale simulation for the analysis of the mechanical behaviors of soft biological tissues was developed and implemented using AMSI. The biotissue simulation is currently a two-scale hierarchical multi-scale simulation coupling two continuum analyses. The primary scale (engineering scale) is a macroscopic simulation using the finite element method on the cauchy momentum balance equation. The tertiary scale is microscopic and takes place at every macro-scale integration point (those points where the stress

response of the material is queried during the macro-scale simulation) and used to supply the material response. The micro-scale simulation simply conducts a force-balance operation on a randomly generated collagen fiber-network residing inside of a unit cube representative volume element (RVE) which is deformed using the state of the associated macro-scale element, and has appropriate boundary conditions enforced. More detail into the derivation of this multi-scale system can be found in [6] [1] [7]. Discussion of the dimensionalization of the dimensionless force terms (the upscaling operation from micro-scale to macro-scale) produced by each fiber network can be found in [8] [3].

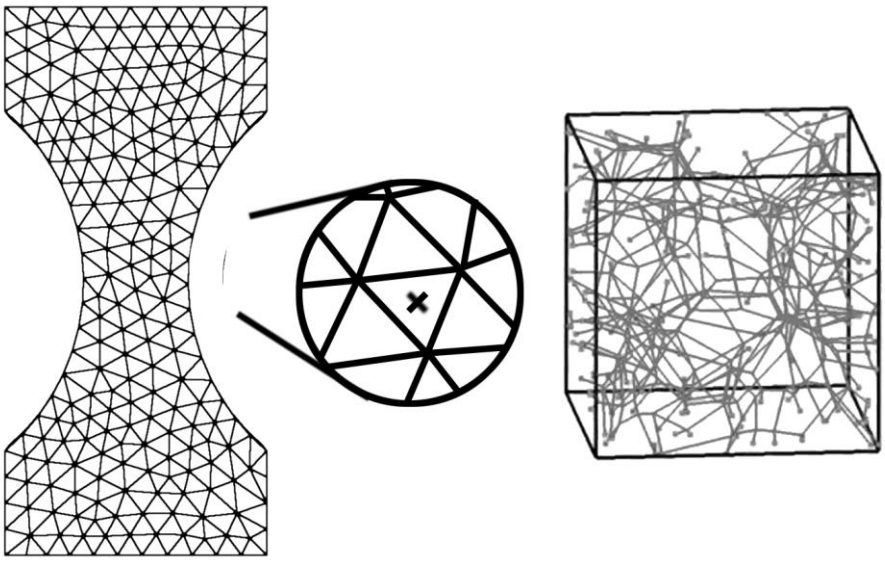

Figure 1: Biotissue multi-scale domains

The biotissue simulation was implemented by combining two single-scale simulations: a typical parallelized finite element analysis and a quasistatics code. The combination of these two simulations using AMSI was very straightforward, involving the introduction of an initialization and configuration phase for AMSI, wrapping the pre-existing main functions of each simulation in an AMSI scale-task callback, and finally introducing relevant AMSI communication (and load-balancing) calls at locations where scale-interaction is required. A visual representation of the biotissue multi-scale relationship can be seen in Figure 1.

The simulation has been run many times on the Amos BlueGene/Q managed by the Center For Computational Innovations at Rennselaer Polytechnic Institute, and has scaled up to 250k elements at macro-scale with over 16k cores (over 90% assigned to micro-scale simulations as these represent the bulk of actual computation). Scaling beyond this should be possible but has not currently been conducted. Theoretically the simulation should be able to scale up to around 300 million elements, limited by memory restrictions on the AMOS machine and the memory associated with a single RVE. Scaling to the full machine is however necessary to get a richer understanding of the performance of AMSI.

Symposia on the development and use of AMSI for the biotissue problem have been given at the SIAM Conference on Parallel Processing for Scientific Computing 2014 and at the SIAM Conference on Computational Science and Engineering 2015.

The Parallel Processing for Scientific Computing talk [11] was a discussion of the general layout and services provided by AMSI and the implementation of the biotissue example problem as well as initial performance results of the biotissue problem on a relatively modest (20k elements) mesh and a more reasonable mesh (200k elements) on the domain seen in Figure 2.

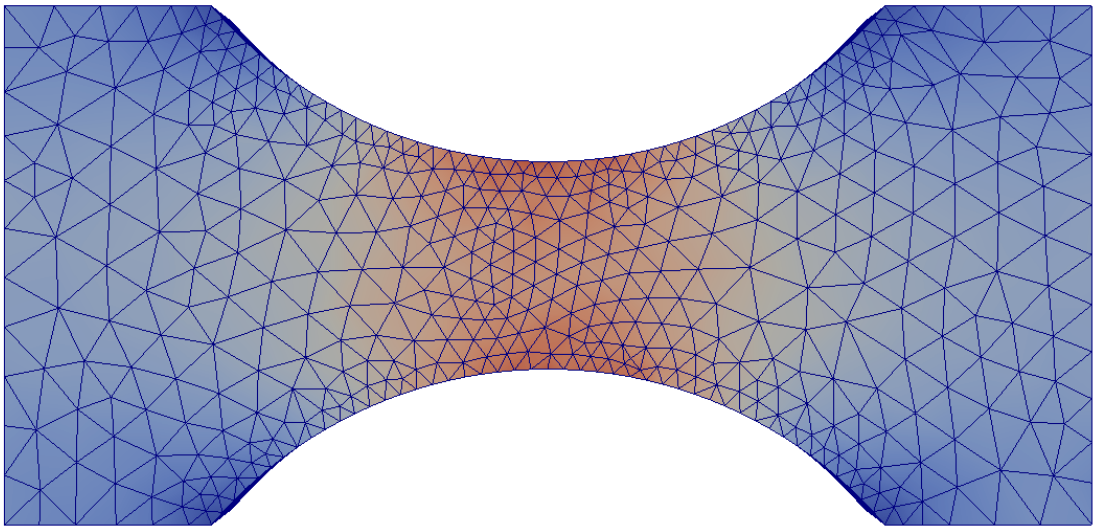

Figure 2: Engineering scale mesh

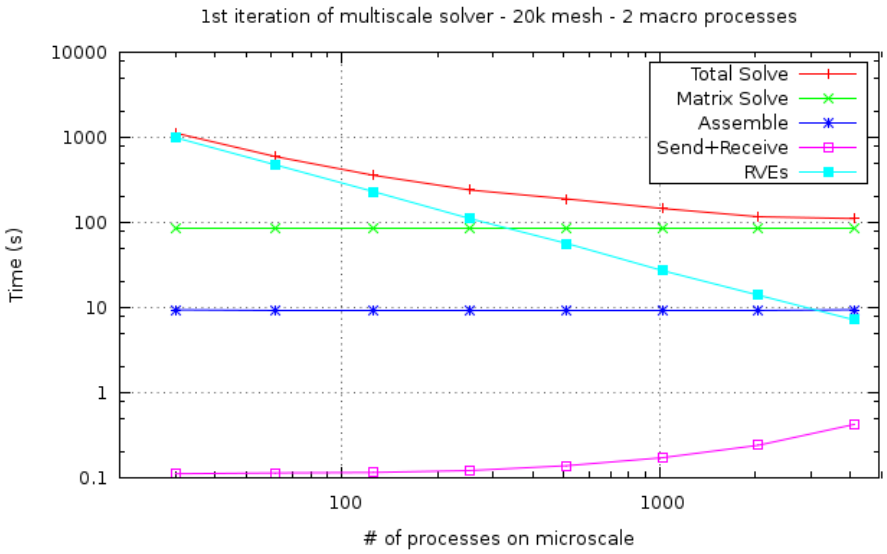

Figure 3: Multi-scale solver 1st iteration (20k elements)

Figure 3 is a strong scaling study conducted on a 20k element mesh, leaving the number of macro-scale processes constant and varying the number of micro-scale processes from just 32 to 8192 by powers of 2. The total time to assemble and solve a single tangent stiffness matrix at the engineering scale is considered (1 Newton-Raphson iteration), and different portions of the overall workload are shown. The actual matrix solve is unchanging as the number of macro-scale processes which compute the linear solve remains constant. Thus the matrix solve time represents the idealized optimal time to compute a single iteration. It is clear that as the number of micro-scale processes increases, the overall solve time approaches the matrix solve time, and while the communication time increases, it is still several orders of magnitude less than the overall solve time.

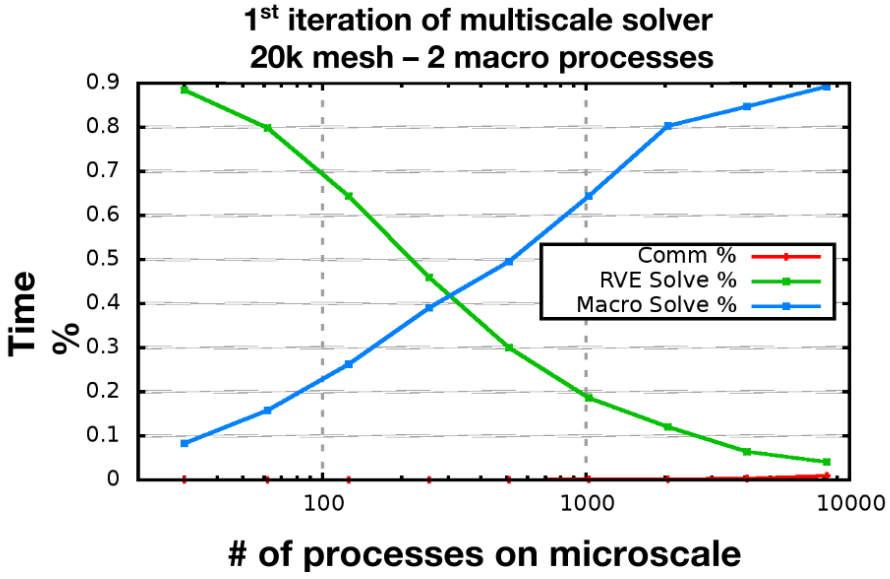

Figure 4: Strong scaling on 20k mesh

Figure 4 is another interpretation of the same strong-scaling study, weighted by the total time-to-solution for each case. It shows that as the number of micro-scale processes increase sufficiently, the overall computational load is dominated by the matrix solve. Increasing the number of processes at macro-scale will result in the time-to-solution for the matrix solve reducing as well, but both the size of macro- and micro-scales (and their ratio) must be considered when trying to determine the optimal allocation size and scale-task assignment. This figure suggests that a 1:128 macro-to-micro ratio will result in equal time being spent on both computations, and allow local load balancing operations at individual scales to make the most impact. Of course overall time-to-solution can be minimized by simply allocating as many processes as are available to solve the problem, but allocating in the above ratio, weighted by the efficacy of individual scale load-balancing to impact the performance of a single scale, should result in better time-to-solution for the multi-scale simulation. Further discussion and analysis of this concept will be discussed in [10].

The Computational Science and Engineering talk [9] was a discussion of the scale-sensitive load-

balancing operations provided by AMSI and their usage in the biotissue example problem. For this talk only the 20k mesh was used as the run times for a 200k mesh proved too long for a reliable turnaround on results, however the longer the computation (in terms of incremental load steps applied) the more benefit should be seen from the scale-sensitive load balancing. Longer runs on a larger (250k element) mesh are currently being conducted, as recent work on performance and memory footprint has allowed much quicker overall execution times making larger runs feasible.

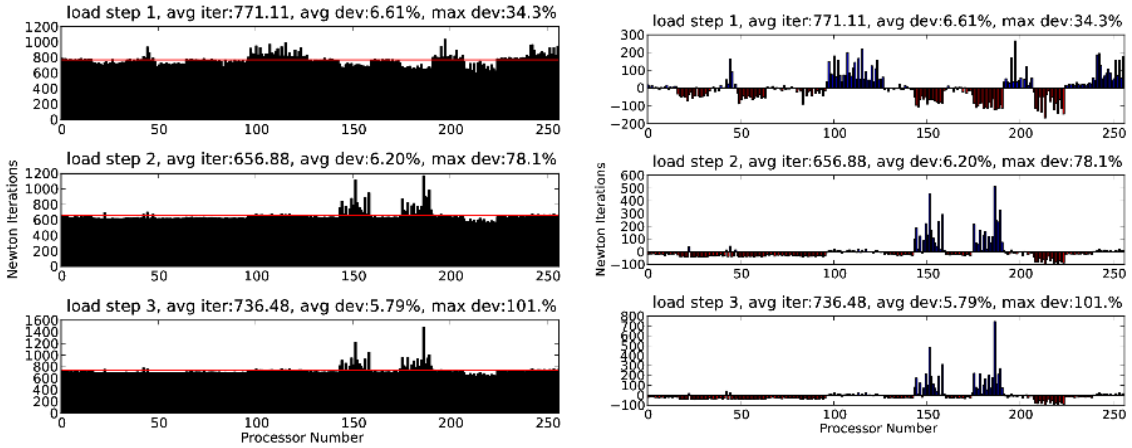

Figure 5: Unbalanced load distribution (20k elements)

As each RVE is a nonlinear simulation there is no a-prior estimate of the computational load each RVE represents, so the optimal initial distribution is simply treating the weight of each RVE as 1. Thus each micro-scale process is assigned either $floor\left(\frac{\text{numRVEs}}{\text{numProcesses}}\right)$ or $ceil\left(\frac{\text{numRVEs}}{\text{numProcesses}}\right)$. Once the first load-balancing event is reached, regardless of the granularity at which load-balancing is occurring for a given case, each RVE has recorded the total number of newton iterations it has conducted since the previous load-balancing event. This number is used as a heuristic weighting metric to determine the computational load of each individual RVE. The iteration count metric was chosen as each iteration takes a similar level of work to complete, up to the difference in the number of RVEs for each unique fiber network. Further, assuming linear incremental loading, regions in the engineering-scale mesh located near stress concentrators will result in 'more interesting' boundary conditions for the micro-scale problems related to numerical integration points in that region, which may result in the RVE requiring more Newton iterations to converge to an adequate solution, since the boundary conditions may lie farther from the initial state of the RVE in the regime of convergence for the nonlinear problem.

The initial distribution of weights is shown in figure 5, with a 20k element macro-scale mesh and 256 processes assigned to compute micro-scale RVEs. The left graph shows the overall weight (the sum of all newton iterations for all RVEs assigned to a given process) while the right graph shows the deviation from the average. While the average deviation is under 10% in all cases, the max deviation is what truly matters since all micro-scale RVEs must finish processing before

scale-coupling communication can take place and the macro-scale can continue to compute the elemental matrices and assemble a global tangent stiffness matrix.

Figure 6 shows the result after scale-sensitive load-balancing (using the Zoltan libraries [4] [2] to plan the actual migration) is conducted. The maximal deviation after the first iteration is unfortunately much higher. But after that first iteration the max deviation is reduced by approximately 80%. This is illustrative of the trend seen in the biotissue simulation, the number of micro-scale newton iterations conducted during the first incremental load step are not illustrative of the number of micro-scale newton iterations required during additional incremental load steps, which stay largely the same. Thus the first load balancing operation over corrects the issue, but the second and subsequent load-balancing operations result in better load balancing and run times overall. This can be seen more clearly in figure 7 which shows the weights for 64 micro-scale processes with the same mesh as above over 9 incremental loading steps, the left graph shows the load without balancing and the right shows the load with balancing. The max deviation without load balancing stays in the 20% range, while with load balancing it typically stays under 10%, as low as 2.77% in the 8th loading step.

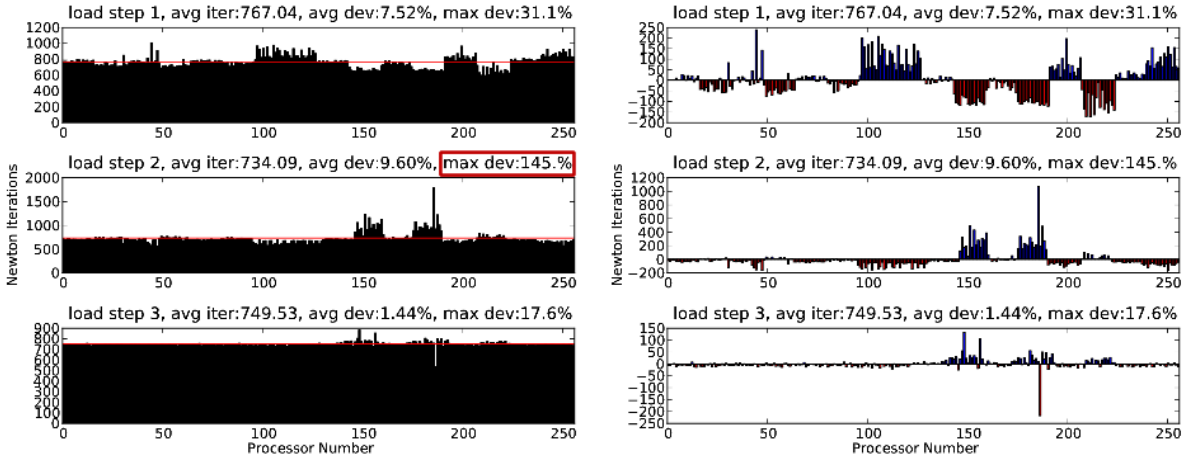

Figure 6: Balanced load distribution (20k elements)

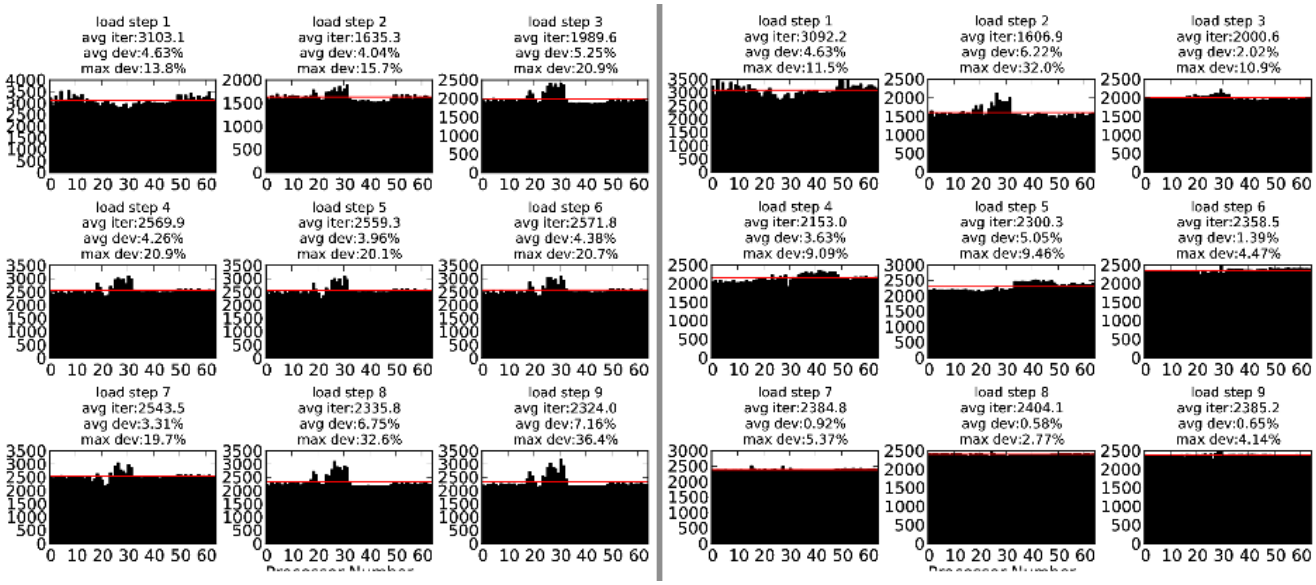

Figure 7: Nine incremental loading steps (20k elements)

ANTICIPATED IMPACT: Currently AMSI supports only static parallel execution space discretization with non-overlapping process sets. The restriction on non-overlapping process sets is currently being worked on to allow temporary scale-reassignment while a scale is blocked waiting on an associate scale and has no work which can be conducted. However, more dynamic management of the execution space represents the most significant current goal of the AMSI project. Implementation of this feature – which will allow scale- tasks to grow and shrink as computational demands associated with the scale change – is dependent on much if not all of the proceeding functionality (especially load balancing and migration features, as all relevant scale data from a process to be dynamically reassigned scales must be recovered and redistributed throughout the related scale).

Scale-balancing is the balancing of scale-tasks against each other to minimize rendezvous time and maximize time spent in useful computation. Scale-balancing is a difficult prospect as it involves man- aging many dynamically-changing scales which themselves are being load-balanced using scale-sensitive load-balancing techniques, and represents a significant and promising area of future work. Once the ability to dynamically manage the process set associated with a scale is developed, algorithms must be developed to make the best use of this feature. Developing these algorithms represents a major - and daunting - area of future work, as there are numerous constraints that must be taken into account in order to optimize such a scheme.

A second micro-scale simulation, where the collagen fibers are embedded in tissue is currently implemented as a standalone simulation. Work to incorporate this into the biotissue example simulation will be conducted when the correct scale-coupling terms have been developed (inclusion in the AMSI system is already accomplished, but the up- and down-scaling operations have not been developed yet so this RVE is not used). [5] [13] [14]

At present AMSI has no meta-model of the computational domain of a problem, or of the relation between the domains of associated scales, it simply knows that the relationship exists. A richer model of the domain relation between problems would likely allow for a wider range of automation in scale-linking and dynamic scale management operations. This will likely not be addressed until a test case where such functionality is needed or beneficial is considered.

An article focusing on the implementation and use of the scale-sensitive load-balancing operations in AMSI for the biotissue example problem is being finalized and will be submitted for publication soon [10]. Additional articles focused primarily on the usage of the Biotissue example simulation to solve novel problems of interest is also being worked on and should see submission in the coming months.

At present there are no current proposals associated with this work for ongoing funding, simply because funding for associated projects (biotissue and other multi-scale projects) has already been secured. Thus work on AMSI will continue during the work on projects which make use of the features it provides. Several interesting multi-scale simulations have been suggested, however, particularly a combination of a Monte-Carlo code and a Magnetostatics code by a fellow researcher. At present this idea is in a preliminary stage of development, though using AMSI would be relatively straightforward and achieving a working code (given two working single-scale codes, and the development of physically-sound scale-linking operations) would likely take only several days of implementation time.

There are several ongoing projects at the Scientific Computation Research Center (SCOREC) on the RPI campus that AMSI will be used with. Particularly SCOREC's ongoing work on incorporating our dynamic mesh operations with the Albany simulation system developed at Sandia National Labs. For multi-scale simulations using the Albany simulation code, we will make use of AMSI's capabilities to be minimally intrusive and only require modifications at those locations in code where scale-coupling terms are needed.

CONCLUSION: The development of the Adaptive Multi-scale Simulation Infrastructure has produced an initial set of tools capable of easing the implementation and execution of multi-scale simulations by combining robust legacy codes in a minimally intrusive manner. The AMSI libraries operate independently of the underlying simulation codes, both in terms of language of implementation and the numerics operations undertaken to achieve solution.

This freedom allows AMSI to be used for a wide variety of multi-scale cases. However this freedom also means that it is possible to do things poorly, as with any sufficiently low-level system. Thus future developments are likely to be focused on building a hierarchy of abstractions atop the currently implemented low-level systems ease in the use of AMSI for the widest-used subset of the cases AMSI provides support for.

A set of operations for managing adaptive multi-scale simulations has been developed and implemented in AMSI, specifically support for dynamically adding and removing scale-tasks at run time. This support is being built on and enriched to allow this complex operation to happen

with minimal programmer intervention, at present the support is low-level and some simulation developer intervention in code to manage the newly-initialized or removed scales is needed. Building on this operation and expanding the metadata model AMSI maintains of the simulation state will allow for the implementation of fully dynamic simulation scale control: arbitrary initialization and destruction of non-primary scales, and allowing blocked process scales to be reassigned (temporarily or permanently) to higher-priority scales.

Additionally, support for load balancing an individual scale of a multi-scale simulation while maintaining the overall integrity of the multi-scale coupling operations has been developed. This allows more traditional load balancing operations – vital to optimizing parallel performance and time-to-solution in traditional massively parallel simulations – to be used on scale codes being used with the AMSI systems. This also represents an important step in the development of general adaptive capabilities for multi-scale simulations, where they greatest challenge is balancing the execution of many interacting (and possibly blocking) simulation scales.

REFERENCES:

. [1] B. Agoram and V. H. Barocas. Coupled macroscopic and microscopic scale modeling of fibrillar tissues and tissue equivalents. Journal of biomechanical engineering, 123(4):362–369, 2001.

. [2] E. G. Boman, U. V. Catalyurek, C. Chevalier, and K. D. Devine. The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing: Partitioning, ordering, and coloring. Scientific Programming, 20(2):129–150, 2012.

. [3] P. L. Chandran and V. H. Barocas. Deterministic material-based averaging theory model of collagen gel micromechanics. Journal of biomechanical engineering, 129(2):137–147, 2007.

. [4] K. Devine, E. Boman, R. Heaphy, B. Hendrickson, and C. Vaughan. Zoltan data management services for parallel dynamic applications. Computing in Science and Engineering, 4(2):90–97, 2002.

. [5] S. P. Lake, M. F. Hadi, V. K. Lai, and V. H. Barocas. Mechanics of a fiber network within a non-fibrillar matrix: model and comparison with collagen-agarose co-gels. Annals of biomedical engineering, 40(10):2111–2121, 2012.

. [6] T. Stylianopoulos. Multiscale mechanical modeling of soft biological tissues. PhD thesis, University of Minnesota, 2008.

. [7] T. Stylianopoulos and V. H. Barocas. Multiscale, structure-based modeling for the elastic mechan- ical behavior of arterial walls. Journal of biomechanical engineering, 129(4):611–618, 2007.

. [8] T. Stylianopoulos and V. H. Barocas. Volume-averaging theory for the study of the

mechanics of collagen networks. Computer methods in applied mechanics and engineering, 196(31):2981–2990, 2007.

. [9] W. R. Tobin, D. Fovargue, V. W. L. Chan, and M. S. Shephard. Load balancing multiscale simulations. SIAM Conference on Computational Science and Engineering, 2015.

. [10] W. R. Tobin, D. Fovargue, V. W. L. Chan, and M. S. Shephard. Scale-sensitive load balancing for numerically multi-scale simulations. in prep.

. [11] W. R. Tobin, D. Fovargue, and M. S. Shephard. Parallel infrastructure for multiscale simulation. SIAM Conference on Parallel Processing for Scientific Computing, 2014.

. [12] E. Weinan, B. Engquist, et al. The heterognous multiscale methods. Communications in Mathe- matical Sciences, 1(1):87–132, 2003.

. [13] L. Zhang, S. Lake, V. Barocas, M. Shephard, and R. Picu. Cross-linked fiber network embedded in an elastic matrix. Soft matter, 9(28):6398–6405, 2013.

. [14] L. Zhang, S. P. Lake, V. K. Lai, C. R. Picu, V. H. Barocas, and M. S. Shephard. A coupled fiber-matrix model demonstrates highly inhomogeneous microstructural interactions in soft tissues under tensile load. Journal of biomechanical engineering, 135(1):011008, 2013.